Globality–Locality-Based Consistent Discriminant Feature Ensemble for Multicamera Tracking

Kanishka Nithin and François Brémond

ALL QUERIES ARE ANSWERED IN " Author Queries" PAGE AFTER PAGE NO 10.

***Abstract*— Spatiotemporal data association and fusion is a well-known NP-hard problem even in a small number of cameras and frames. Although it is difficult to be tractable, solving them is pivot for tracking in a multicamera network. Most approaches model association maladaptively toward properties and contents of video, and hence they produce suboptimal associations and association errors propagate over time to adversely affect fusion. In this paper, we present an online multicamera multitarget tracking framework that performs adaptive tracklet correspondence by analyzing and understanding contents and properties of video. Unlike other methods that work only on synchronous videos, our approach uses dynamic time warping to establish correspondence even if videos have linear or nonlinear time asynchronous relationship. Association is a two-stage process based on geometric and appearance descriptor space ranked by their inter- and intra-camera consistency and discriminancy. Fusion is reinforced by weighting the associated tracklets with a confidence score calculated using reliability of individual camera tracklets. Our robust ranking and election learning algorithm dynamically selects appropriate features for any given video. Our method establishes that, given the right ensemble of features, even computationally efficient optimization yields better accuracy in tracking over time and provides faster convergence that is suitable for real-time application. For evaluation on RGB, we benchmark on multiple sequences in PETS 2009 and we achieve performance that is on par with the state of the art. For evaluating on RGB-D, we built a new data set.**

***Index Terms*— XXXXX.**

I. Introduction

THE goal of this paper is to: 1) provide a real-time solution with good accuracy to estimate states of multiple targets relative to its complement in multicamera environment and 2) conserve the identities of targets and produce unfragmented long trajectories under variations in appearance and motion over time. In spite of the number of solutions, real-time multitarget tracking across multiple camera network with reasonable overlap is still considered most challenging and unsolved computer vision problem. This is mainly due to placement of cameras, time asynchronous cameras, multicamera calibration, distortions, parallelism, fuzzy data association, and fusion across network of cameras. Despite challenges, multicamera systems are crucial because they help in obtaining more visual information about the same scene that complements each other, thereby helping in overcoming traditional deficits of single-camera object tracking and improving higher vision tasks such as activity recognition and surveillance.

Offline and global association methods usually require detection and tracking results for entire sequence prior to data association. This leads to high computation due to iterative associations across multiple cameras for generalizing globally optimized tracklet association and fusion; therefore, they are difficult to apply for real-time applications. Global approaches are also more exposed to local optima solutions compared with online methods, whereas our method performs online associations and fusion based on optimal frame buffer containing the information gathered till the present frame. Hence, our approach reduces the ambiguity in global associations and it produces competing performance to the state of the art while being suitable for real-time applications. As a byproduct, shortcomings of online frame buffer-based tracking are implicitly overcome by multicamera system setup.

Unlike some of works mentioned in Section II, the proposed online multicamera tracklet association is designed considering two key criteria—inter- and intra-camera consistency and discriminability of trajectory features. Our method incrementally learns and updates the discriminative appearance model belonging to each trajectory and ranks them based on consistency and discriminancy of the candidate tracklets. We also use 3D projected geometric information in conjunction with long-term appearance features for efficient data association even in challenging situations.

In our approach, we use planar homography to establish 3D common referential between cameras onto which the 3D points of each tracklet from all cameras are projected. Dynamic time warping (DTW) algorithm is used to find one-to-one frame mapping between linear or nonlinear time asynchronous cameras. DTW also selects candidate tracklets for association. Tracklet association is modeled as a sequence of complete bipartite graphs. Association score for each pair of tracklets is calculated as ensemble of geometric and appearance features weighted by globality–locality consistent discriminant score (GLCDS). GLCDS is learnt as an estimate of discriminancy weighted consistency score. Discriminancy of individual features is calculated as fisher score of that feature over entire tracklets. Consistency of each feature is calculated as deviation

Manuscript received December 16, 2015; revised April 27, 2016 and August 10, 2016; accepted September 22, 2016. This work was supported in part by the Agence Nationale de la recherche under Grant ANR-13-SECU-0005-01 of the Project MOVEMENT and in part by the Program COSG 2013. This paper was recommended by Associate Editor H. Yao.

The authors are with INRIA Sophia Antipolis Méditerranée, 06902 Sophia Antipolis, France (e-mail: kanishka-nithin.dhandapani@inria.fr; francois.bremond@inria.fr).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCSVT.2016.2615538

of that feature over a distribution belonging to the tracklet under consideration. Fusion is performed using confidence score-based adaptive weighting method. This enables correct and consistent trajectory association and fusion even if the individual trajectories have inherent noises, occlusion, and false positives.

Our method has the following advantages.

1) We integrated measures that account properties, nature of video, and its contents for online feature selection and combination. It automatically elects the best feature ensemble based on the video contents and properties.
2) We lack real-time state-of-the-art approaches in multi-camera tracking. This is attributed to the heavy optimizers used in such approaches. Our method reduces the burden of relying on such heavy optimizers by concentrating on feature engineering. Our approach produces state-of-the-art comparable performance in real time by avoiding computationally expensive optimization, metrics, and data-gathering (fusion) strategy, thus significantly influencing on the scalability of network as well.
3) Our cost function allows us to efficiently model multilevel relationship among tracklets such as a spread of global, local, and motion features used in our method.
4) Our approach leverages depth information upon availability to complement RGB data to overcome shortcomings of RGB cameras and other issues like privacy.

The reminder of this paper is divided into the following sections. In Section II, we review some significant previous work and how our method differs from them. In Section III, we review multicamera synchronization and multiview geometry used in our approach. Next, in Section IV, we discuss how we formulate trajectory association problem, followed by Section V that describes calculation of trajectory similarity metrics. Section VI briefs on consistency and discriminancy of cross-view tracklets and GLCDS calculation. Trajectory fusion is introduced in Section VII, the experimental results are presented in Section VIII, and finally, Section IX concludes this paper.

II. Related Work

In recent years, there have been comparatively less multicamera data association and tracking approaches proposed. Most of the multicamera approaches in recent times have concentrated mainly on offline approaches. On a general basis, approaches can be outlined based on: 1) fusion time—either early fusion [2] or late fusion [3] and 2) the search space—greedy, i.e., temporally local (online) or global optimization with longer temporal stride (offline) [4], [5].

Approach [1] extends the work of [6] to jointly model multicamera reconstruction and global temporal data association using MAP. They use global min cost flow graph for tracking across multiple cameras. Berclaz *et al.* [6] have detection based on probability occupancy map. They also use flow graph-based method for solving both mono-camera and multicamera setup within a restricted and predetermined area of interest. The drawback of such min cost flow graphs that currently own the state of the art is that they are not real time as the complexity increases with more cameras in the network since combinations of observations from multiple cameras increase exponentially and the costs need to be predefined. Min-flow graphs cannot work with higher order motion models as their cost function cannot be factored into product or sum of edges of adjacent nodes. Reference [19] solves the association problem by first solving 3D hypothesis from multiple camera object detection fusion and then by solving temporal data association. The drawback is unnecessary overhead where the problem is diversified into two separate problems of 3D reconstruction fusion at central server and solving to assign back the reconstructed fusion into 3D tracklets established by individual sensors.

Evans *et al.* [7] use early fusion strategy for detection inspired from [2] and extend it for multicamera tracking and estimating object size in multicamera environment. Their approach leverages multiview information into early stage (detection) of pipeline to remove ghosts. Since the synergy map they use for ghost suppression also suppresses existing objects in the previous frame, they cannot perform tracking by associating detections moment to moment. Multivariate optimization is performed on object size together with probable location of object in the next frame. The objective function involves both object size estimate and tracking information, and the solution may be suboptimal and is not real time. By nature of their ghost suppression method that involves intricate assumptions such as line of view from camera to object assumptions, it makes it difficult to track objects in cluttered or crowded environment.

Anjum *et al.* [8] have presented an unsupervised inter-camera trajectory correspondence algorithm. For the association step, they propose a hybrid approach: project the trajectories from each camera view to the ground plane in order to find associations among trajectories, and then, make image-plane reprojections of the matched trajectories. These methods rely entirely on goodness of homography, smallest margin of error in calibration gets added up during initial projections and reprojections. Thus, these methods are susceptible to introduce errors that end up being association errors. Sheikh *et al.* [9] have proposed a target association algorithm that addressed the problem of associating trajectories across multiple moving airborne cameras with a constraint that at least one object is seen simultaneously between every pair of cameras for at least five frames. Since this method uses object centroid as feature points to recover the homography and later uses RANSAC to find out best subset of such points to find correspondence, it works well when in sparse environment, but in dense environment, it may fail. Their approach assumes that all the objects to be tracked are on the common ground well aligned with all the cameras present in the network.

To address the shortcomings of the methods discussed above, we propose a framework that synthesizes local feature level information into the global object level based on consistent discriminant election and weighting for multitarget tracking.

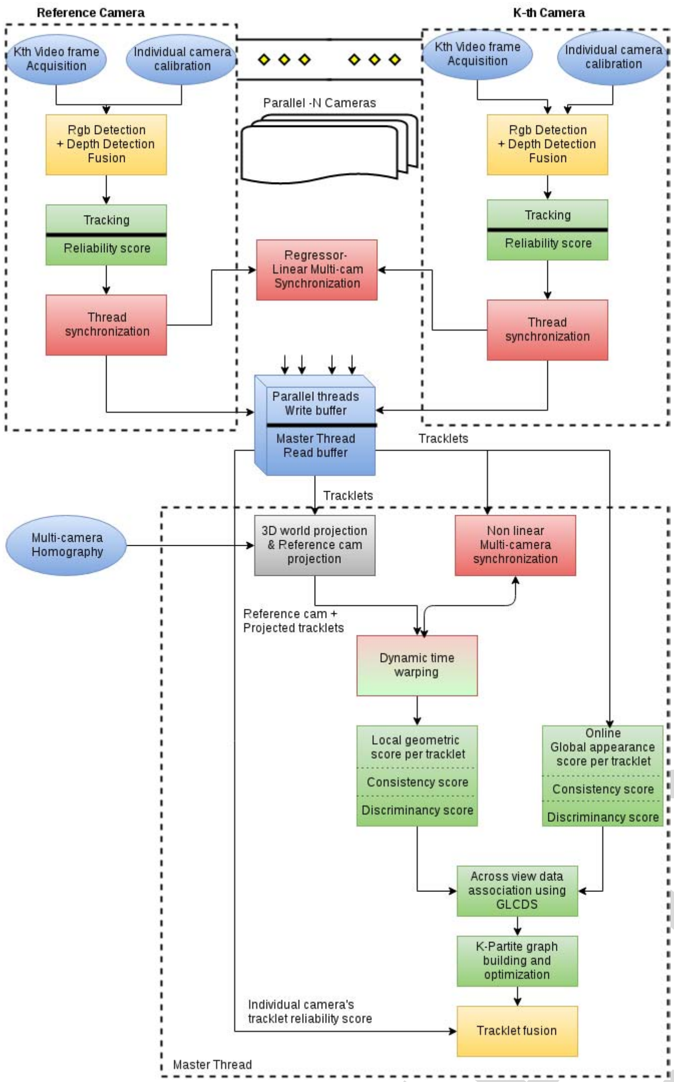

Fig. 1. Pipeline of our approach.

III. Proposed Methodology

General system architecture and pipeline can be seen in Fig. 1. Multiple worker threads process each camera in a network at the same time. All threads perform detection and tracking as independent worker nodes. After a buffer time, these threads synchronize to push their data onto the master thread where all key multicamera-related work is done, i.e., building online tracklet appearance models, local features, 3D projection, online learnt feature ensemble, association, and fusion.

IV. Multicamera Synchronization and Multiview Geometry

Elementary and most key settings for our multicamera tracking system are as follows.

1) The cameras in the network need to be time synchronized with respect to reference camera C^{ref}. Here by reference camera, we mean a chosen camera onto which the geometric data from other cameras in a network are projected to.
2) Individual camera calibration for projection onto a 3D world W^{Ck} belonging to that camera.
3) Multiview homography that establishes a mapping between world of camera k W^{Ck} and world of reference camera W^{ref}

Most of previous approaches assume that cameras are time synchronized, but we also handle the case of linear and nonlinear asynchronization between the cameras. If the cameras are linearly asynchronous, we need to map each frame in camera C^k to corresponding frame in reference camera C^{ref}. We accomplish this task using linear regression. Given a set of values, the linear regression model assumes that the relation between the dependent variable F^{C^k} and T^{C^k} variable is linear. F^{C^k} are frames from camera k, and T^{C^k} are timestamps T from camera k. The relation between both variables can be approximated as linear as

$$F^{C^k} = t_0^{C^k} + \text{slope}^{C^k} \times T^{C^k} \tag{1}$$

where C^k is the kth camera. For simplicity, we assume constant $t_0^{C^k} = 0$.

In order to find a relation between each video, we can equate the timestamps of both cameras $T^{C^k} = T^{C^{\text{ref}}}$

$$T^{C^k} = \frac{F^{C^k}}{\text{slope}^{C^k}} = T^{C^{\text{ref}}} = \frac{F^{C^{\text{ref}}}}{\text{slope}^{C^{\text{ref}}}}. \tag{2}$$

After if we know the parameters slope^{C^k} and $\text{slope}^{C^{\text{ref}}}$, we can map from the frame of one camera to the other. This parameter can be obtained from expressions

$$\text{slope}^{C^k} = \frac{\Delta F^{C^k}}{\Delta T^{C^k}}. \tag{3}$$

Then the camera with lower frame rate is taken as reference, and the synchronization for the camera C^k is calculated as

$$F^{C^k} = \frac{\text{slope}^{C^k}}{\text{slope}^{C^{\text{ref}}}} \times F^{C^{\text{ref}}}. \tag{4}$$

If the cameras are nonlinearly asynchronized, we use DTW as a way to establish approximate frame-to-frame correspondence between them. Here DTW also doubles as a dynamic programming approach to speed up the process of finding geometric similarity between the tracklets that need to be associated. More details on DTW and the process are explained in Section V-A.

A moving person viewed from different points of view results in different trajectories. The estimation of the homography between these views is the key in establishing association between them. Our multiview calibration is based on planar homography.

Points projected on a 3D world W^{Ck} from the kth view may be related to the corresponding image points in the 3D world W^{ref} in reference view using planar homography. The idea is to project the trajectory points from all cameras under consideration onto the common referential world. In our case, common referential is reference camera coordinate system. Given a point X in the kth view, the problem consists in finding

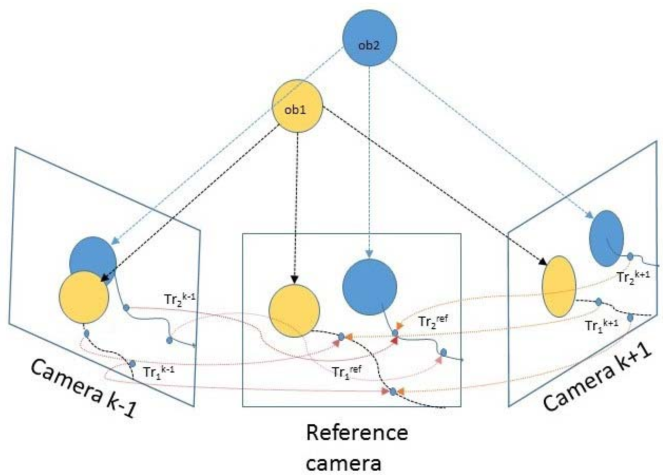

Fig. 2. Projective transformation of tracklet points belonging to Tr_1^{k-1}, Tr_2^{k-1} between images from camera $k-1$ and image plane of reference camera using homography H induced by that plane. The same happens with camera $k+1$ and so on.

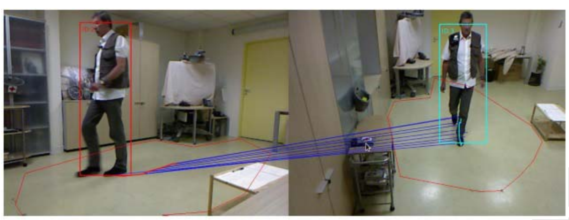

Fig. 3. Corresponding projections on reference image plane. Left: reference image plane. Blue lines represent the projection of points from nonreference camera to the image plane belonging to reference camera.

the corresponding point X' in the reference view. The relation between the first and the second view is given by

$$X' = H_\pi \cdot X. \tag{5}$$

Once we found the homography between views, we can project the trajectories from one camera view to the other one as shown in Figs. 2 and 3.

V. Multiview Trajectory Association

Generalized maximum /minimum clique problem or K-partite problem, where finding the clique with maximum score or minimum cost is an NP-hard problem as shown in [20]. Since there is no polynomial time solution to this problem, we breakdown the problem by reducing it to sequential bipartite matching problem between reference camera and any other camera C^k in the network. Let us say we have K cameras $\{C^{\text{ref}}, C^1, C^2 \ldots C^k\}$, and we reproject all the trajectories from cameras $\{C^1, C^2..C^k\}$ to reference camera C^{ref} and perform trajectory association, similarity calculation on C^{ref}. The associated tracklets between the reference camera and the kth camera are accumulated until tracklet associations for all $\{C^{\text{ref}}, C^k\}$ pairs are solved. Once all the tracklet associations from each camera pair are available, the fusion is done in the reference camera C^{ref}. By doing this way, it leads to estimation of optimal solution for NP hard problem in polynomial time.

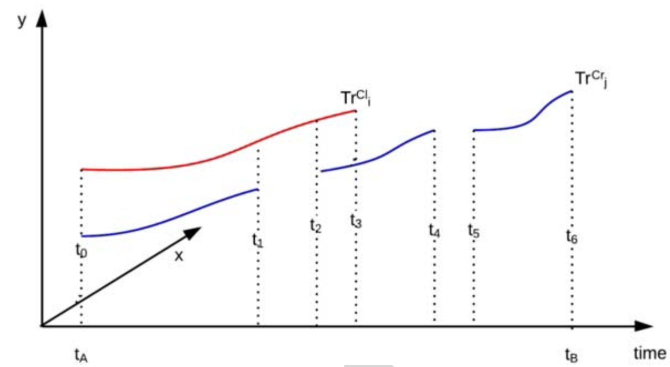

Fig. 4. Two tracklets in common subintervals between two cameras in the time interval $[t_A, t_B]$.

The association problem in general is related to the need of establishing correspondences between pairwise similar trajectories that come from different overlapping cameras.

The association or correspondence may be modeled as a sequence of bipartite graph matching problem in which each set S_k has trajectories that belong to camera k. For example, for a reference camera C^{ref} and any other overlapping camera C^k, a set of trajectories S_{ref} and S_k is defined.

A bipartite graph is a graph G in which the vertex set V can be divided into two disjoint subsets S_{ref} and S_k such that every edge $e \in E$ has one end point in S_{ref} and the other end point in S_k. Each object being tracked is denoted by TO_i in the resulting observation (i.e., a track point) of the multitarget tracking algorithm. The tracked objects have been synchronized in terms of frame number F, and they have 2D space coordinates (x, y). Thus

$$\mathrm{TO}_t = (F, (x, y))_t.$$

Let TO^i represent the ith tracked object that belongs to the trajectory $\mathrm{Tr}_i^{C^k}$ observed in the camera C^k where $k = l, r$. Thus, each trajectory is composed by a time sequence of 3D points of physical objects

$$\mathrm{Tr}_i^{C^k} = \left\{\mathrm{TO}_0^i, \mathrm{TO}_1^i, \mathrm{TO}_t^i, \ldots, \mathrm{TO}_{n_i}^i\right\} \tag{6}$$

where n_i is the length of the above trajectory. Consequently, each camera C^k has a set of N and M trajectories belonging to sets S_{ref} and S_k

$$S_{\text{ref}} = \left\{\mathrm{Tr}_0^{C^{\text{ref}}}, \mathrm{Tr}_1^{C^{\text{ref}}}, \mathrm{Tr}_2^{C^{\text{ref}}}, \ldots \mathrm{Tr}_N^{C^{\text{ref}}}\right\} \tag{7}$$

$$S_k = \left\{\mathrm{Tr}_0^{C^k}, \mathrm{Tr}_1^{C^k}, \mathrm{Tr}_2^{C^k}, \ldots \mathrm{Tr}_M^{C^k}\right\}. \tag{8}$$

We abstract the trajectory association problem across multiple cameras as follows. Each trajectory $\mathrm{Tr}_j^{C^k}$ is a node of the bipartite graph that belongs to the set S_k linked with the camera C^k. A hypothesized association between two trajectories is represented by an edge in the bipartite graph. The goal is to find the best match in the graph.

A. Time Overlapping Trajectories

For each hypothetical association, we first filter and remove the associations of trajectories that do not overlap in time.

In the case of time overlapping trajectories, we take the intersecting time interval between them, that is, the lower and the highest time value between both trajectories to get a new time interval in which both trajectories are contained. In the example of Fig. 4, we have two trajectories $\mathrm{Tr}_i^{C_l} \in S_l$ with $0 < i < N$ and $\mathrm{Tr}_j^{C_r} \in S_r$ with $0 < j < M$, and the resulting overlapping time interval is $\Delta t = [Tr^{C^l}(t_0), Tr^{C^r}(t_f)]$. In order to apply DTW, we need trajectories of the same size to be compared frame by frame. The gaps or missing points (due to miss detections or occlusions) are completed with local linear interpolation and smoothing for the mentioned time interval Δt.

B. Linear Interpolation and Smoothing

Object detection is not perfect due to occlusions, visibility, density of crowd, and placement of camera, and thus, a linear interpolation is applied in order to reach a more complete trajectory. We assume that a person follows uniform linear motion between the next and the previous frame. Based on that, a linear interpolation is performed in order to correct miss detections of time length equal to Δ frame(s) at a time. In our experiments, we heuristically limit usage of interpolation up to $\Delta = 4$, and more than four missing detections would be treated as disappearance of object. To perform this correction, position of the person in the current frame is estimated as

$$\mathrm{Tr}_i^{C^k}(t) = \frac{\mathrm{Tr}_i^{C^k}(t-1) - \mathrm{Tr}_i^{C^k}(t+\Delta)}{\Delta} \tag{9}$$

where Δ is the difference between the previous and the next available detection's frame number. $\mathrm{Tr}_i^{C^k}(t)$ is the position of tracked object at time t, $\mathrm{Tr}_i^{C^k}(t-1)$ is the position of tracked object at time $(t-1)$, $\mathrm{Tr}_i^{C^k}(t+\Delta)$ is the position of tracked object at time $(t+\Delta)$, and C^k is the camera number.

The 2D space of the trajectories that belongs to the kth camera is projected to 2D space of ref camera in order to compare and find similar trajectories. During this task, some noise can arise. Thus, in order to deal with this noise, we smooth the trajectory for better results. At this time, we are almost ready to compute the trajectory similarity. However, the common tracklets between both trajectories need to be found.

C. Find Tracklets in Common Subintervals

Fig. 4 shows a graphic illustration of two overlapping trajectories in time interval $[t_A, t_B]$. The x and y axes correspond to geometric space, i.e., geometric x, y coordinates, and t-axis corresponds to time. The two trajectories have two tracklets in the subintervals $[t_0, t_1], [t_2, t_3] \subset [t_A, t_B]$ belongs to trajectories $\mathrm{Tr}_i^{C^l}$, $\mathrm{Tr}_j^{C^r}$, two tracklets in the subintervals $[t_3, t_4], [t_5, t_6] \subset [t_A, t_B]$ belongs to $\mathrm{Tr}_j^{C^r}$. Finally, one tracklet in $[t_1, t_2] \subset [t_A, t_B]$ belongs to $\mathrm{Tr}_i^{C^l}$.

Later on, a trajectory similarity algorithm is applied for every pair of tracklets in common subintervals among both trajectories. It is important to note that now the tracklets have the same length and have been synchronized.

VI. Trajectory Similarity Calculation

The comparison of two temporal sequences invariant to time and speed (e.g., trajectory) and their similarity measurement is done using DTW. There are several trajectory similarity measurements in the state of the art. Two similarity models draw our attention: longest common subsequence described in [10] and DTW introduced in [11]. Among these, we choose the latter as it offers enhanced robustness, particularly being sensible to noisy data. As our goal is to associate trajectories, we need a local measurement for trajectories' comparison that is being done using DTW.

A. Time-Invariant Tracklet Alignment and Similarity

DTW is a distance measure for measuring similarity between two temporal sequences that may vary in time or speed. DTW-based similarity measure works well between cameras having both linear and nonlinear FPS mapping. As a first step in DTW, we place the trajectories in a grid in order to compare them, and initialize every element as ∞ (represent ∞ distance). Each element of the grid is given by $d(\mathrm{Tr}_i^{C^l}(t_i), \mathrm{Tr}_j^{C^r}(t_j))$ representing Euclidean distance that is the alignment between two trajectories' points $\mathrm{Tr}_i^{C^l}(t_i), \mathrm{Tr}_j^{C^r}(t_j) \forall t_i \in [0...n], \forall t_j \in [0...n]$, where n is the length of the shortest trajectory.

Many paths connecting the beginning and the ending point of the grid can be constructed. The goal of DTW is to find the optimal path that minimizes the global accumulative Euclidean distance between both trajectories of size n

$$D\left(\mathrm{Tr}_i^{C^l}, \mathrm{Tr}_j^{C^r}\right) = \min\left[\sum_{t_i, t_j = 1}^{N} d\left(\mathrm{Tr}_i^{C^l}(t_i), \mathrm{Tr}_j^{C^r}(t_j)\right)\right] \tag{10}$$

$$\begin{aligned} D(n,m) &= d(\mathrm{Tr}_i^{C^l}(n), \mathrm{Tr}_i^{C^r}(m)) \\ &\quad + \min\left\{\begin{matrix} D(n-1,m) \\ D(n-1,m-1) \\ D(n,m-1) \end{matrix}\right\}. \end{aligned} \tag{11}$$

The warping path point predecessor of $D(n,m)$, denoted by α, is selected as the one that gives the smallest accumulative distance of the three neighbors as

$$\alpha(t+1) = \min\left\{\begin{matrix} D(n-1,m) \\ D(n-1,m-1) \\ D(n,m-1) \end{matrix}\right\}. \tag{12}$$

Finally, the optimal warping path is a sequence of accumulative distances from the first element of each trajectory until the end

$$\acute{\alpha} = \alpha(t_0), \alpha(t_1), \ldots \alpha(t_i), \ldots, \alpha(t_N). \tag{13}$$

We can see in Fig. 5 that the tracklets are very similar from frame 65 to 82, but after seem like they start to be unequal. The further close the optimal path wanders around the diagonal, the more the two sequences match together.

We could use the immunity/invariance DTW has for time misalignment in time series sequences while aligning the tracklets from different cameras. We use this property of DTW and try to infer a statistic, which could help us approximate

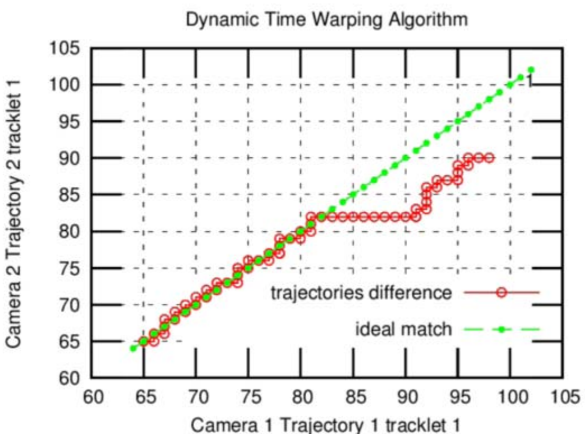

Fig. 5. DTW results for tracklet 1 of two trajectories' comparison. In X and Y, the frames are shown. The optimal path is represented in green, and the DTW result is shown in red.

the nonlinear mapping between certain time asynchronous cameras in network. We process the shape of the DTW warping path (red as shown in Fig. 5) to retrieve information on complementary frame pairs belonging to warping path. In other words, we decode the DTW warping path in terms of frames. The extracted complementary pairs act as one-to-one frame mapping between the cameras under consideration.

VII. Online Learnt Globality–Locality Feature Ensemble

The core idea of our approach is ranking and selection of global–local features to form an ensemble that is crucial for tracklet association while giving good inter-camera discriminability between tracklets. Using only local association information leads to produce shorter fragmented fused trajectories. This may even cause the fusion to drift when one of the cameras has lot of occlusions as it is based on frame-to-frame information. Using only global information leads to more iterative associations as global information induces more confusion. Associations are unreliable when there are lots of distortions existing between cameras. Thus, it is important to strike a balance between these informations while extracting the most consistent and discriminate of them for calculating association. This helps in compensating for the limitations of each feature for a given video.

It is a known fact that feature combinations capture more underlying semantics than single feature patterns. But using less influential pattern combination may not improve the performance of a tracker mainly due to limited discriminability of individual feature. Trajectory similarity is calculated as a two-stage approach (local and global). An ensemble of local and global features is used for determining similarity score. The electing weights that decide the ensemble are learnt online based on the consistency and maximum discriminability of the feature distributions.

A. Local Tracklet Similarity

At local stage, importance is given to local frame-to-frame geometric information. From DTW results, we calculate some statistics like proximity.

Proximity as Euclidean Distance Mean: From DTW results, we calculate normalized pixel Euclidean distance mean for each trajectory comparison and each edge of the bipartite graph. To normalize the DTW results, we divide by the maximum possible distance between both trajectories, that is, the size of the image

$$\mathrm{EDM} = D\left(\mathrm{Tr}_i^{C^l}, \mathrm{Tr}_j^{C^r}\right)/n. \tag{14}$$

B. Global Tracklet Similarity

At global stage, information pertaining to overall appearance of the object throughout the tracklet is taken into account for determining the similarity between tracklets. Feature patterns used for determining an overall appearance score are updated online regularly for the entire trajectory. A global matching score (GMS) quantified from features below represents global tracklet similarity.

Global Matching Score: Appearance-based cues have played a vital role in tracklet association rule mining. Given a set of appearance cues, we create an ensemble of high-quality ones for effective discrimination between tracklet association candidate matches. We extend mono-camera tracklet reliability descriptor work in [12] to suit our approach. We use k=7 cues for our work.

1) *2D Shape Ratio ($k = 1$) and 2D Area ($k = 2$):* Shape ratio and area of an object are obtained from respective bounding boxes, and within a temporal window, they are immune to lighting and contrast changes. Thus, they are one of the good cues to use.
2) *Color Histogram ($k = 3$) and Dominant Color ($k = 4$):* It is basically a normalized RGB color histogram of pixels inside bounding box of moving object. Dominant color descriptor is used to take into consideration only important colors of object.
3) *Color Covariance Descriptor ($k = 5$):* Color covariance descriptor is a covariance matrix that characterizes the appearance of regions in image and is invariant to size and identical shifting of color values. Therefore, color covariance descriptor resists to illumination changes.
4) *Motion Descriptor ($k = 6$):* Depending on the context, constant velocity model or Brownian model is used to describe motion represented by Gaussian distribution. It is useful when objects have a similar appearance.
5) *Occlusion ($K = 7$):* Occlusions significantly degrade the performance of tracking algorithm, and we progressively analyze occlusion by exploiting the spatiotemporal context and overlap information between the tracked object and other objects.

We define tracklet Tr_p as an overlapping tracklet of tracklet Tr_i if tracklet Tr_p has at least one frame overlap with tracklet Tr_i (called as temporal overlap) and the 2D distance of both tracklets is below a predefined threshold (called as spatial overlap). We define tracklet Tr_j as candidate matching tracklet of tracklet Tr_i if it satisfies temporal constraint like the last object detection of Tr_i must appear earlier than the first object detection of Tr_j and a spatial constraint like that the last object detection of Tr_i can reach the first object detection of Tr_j after a number of frames of potential misdetection with the current frame rate.

To ensure reliable tracklet association, [12] weights the discriminative appearance and motion model descriptors and generates a GMS. The GMS of tracklet Tr_i with each tracklet in its matching candidate list (Tr_j) is

$$\mathrm{GMS}\left(\mathrm{Tr}_i^{C^l}, \mathrm{Tr}_j^{C^r}\right) = \frac{\sum_{k=1}^{6} w_k^{ij} \cdot \mathrm{DS}_k\left(\mathrm{Tr}_i^{C^l}, \mathrm{Tr}_j^{C^r}\right)}{\sum_{k=1}^{6} w_k^{ij}} \quad (15)$$

where w_k^{ij} are corresponding weights of each feature descriptors $\mathrm{DS}_k(\mathrm{Tr}_i^{C^l}, \mathrm{Tr}_j^{C^r})$ calculated online by modeling them directly proportional to descriptor similarity of a tracklet with its matching candidate and inversely proportional to descriptor similarity of other overlapping tracklets.

If $(\mathrm{Tr}_i, \mathrm{Tr}_j)$ are matching candidates, $(\mathrm{Tr}_i, \mathrm{Tr}_p)$ are other overlapping tracklets, and their discriminative descriptor weight is calculated as

$$w_k^{i,j} = \zeta^{[\mathrm{DS}_k(\mathrm{Tr}_i, \mathrm{Tr}_j) - \tilde{X}(\mathrm{DS}_k(\mathrm{Tr}_i, \mathrm{Tr}_p)) - 1]} \quad (16)$$

where $\zeta = 10$ determined experimentally and $\tilde{X}$ is the median of the similarities between tracklets $(\mathrm{Tr}_i, \mathrm{Tr}_p)$. The advantage of the median is that its value is not affected by a few of extremely big or small values. The discriminative weight for motion cue alone is calculated as

$$w_6^{i,j} = 0.5 - 0.5 \max_{k=1\ldots5} \left(w_k^{i,j}\right). \quad (17)$$

C. Globality–Locality Consistent Discriminant Score

A cost matrix A is built to represent the cost of association between two tracklets ($\mathrm{Tr}_i^{C^l}$, $\mathrm{Tr}_j^{C^r}$). Each element of such an association cost matrix represents GLCDS weighted sum of Euclidean distance and GMS between the two trajectories. An entry in association cost matrix A can be defined as

$$\begin{aligned} A\left(\mathrm{Tr}_i^{C^l}, \mathrm{Tr}_j^{C^r}\right) &= \lambda_m(\mathrm{Tr}_i) \cdot \mathrm{EDM}\left(\mathrm{Tr}_i^{C^l}, \mathrm{Tr}_j^{C^r}\right) \\ &\quad + (1 - \lambda_m(\mathrm{Tr}_i)) \cdot \mathrm{GMS}\left(\mathrm{Tr}_i^{C^l}, \mathrm{Tr}_j^{C^r}\right) \quad (18) \end{aligned}$$

where λ_m is GLCDS learned to obtain appropriate ensemble feature combination and is discussed further later in Section VI-D.

Now the bipartite graph is complete and the weight W_{ij} of each edge $e \in E$ in $G = (V; E)$ is $A(\mathrm{Tr}_i^{C^l}, \mathrm{Tr}_j^{C^r})$ given by (18).

λ_m helps to decide a tradeoff between local information extracted from frames or global appearance information from tracklets. The learnt weight helps in better feature selection and combination to enhance inter-tracklet discrimination and also cope up with intra-tracklet variations. In this approach, both local geometric and global appearance feature patterns complement each other and are impactful in situations where the data set involves significant appearance changes across object pose, illumination, viewing angle, and different camera parameters.

1) Color Calibration Across Cameras: To calculate consistency and discriminative power of tracklet features across cameras, we need to color calibrate the cameras for accounting color distortion between them. Therefore, as a preprocessing step before validating discriminability and consistency, we perform histogram specification and histogram matching, i.e., we project and transform the histogram of any camera C^k onto histogram of reference camera C^{ref}. Level of color distortion after specification is validated by comparing the transformed histogram and reference histogram using correlation-based histogram matching.

Even if appearance model of a tracklet is discriminative, it makes sense to weight them high only if the features in the model are consistent and vice versa. Thus, λ_m is calculated as *an estimate of discriminant score weighted consistency of individual features*.

D. Discriminative Power of Tracklet Features

Discriminative power of the GMS features is calculated as a mean of normalized fisher scores of individual GMS tracklet features. Fisher score is a quantitative measure popularly used in statistics for numerically solving maximum likelihood problems. In computer vision, fisher score is used to rank the best set of features, such that in the space spanned by selected features, the distances between datapoints of different classes are as large as possible, while distances between datapoints of the same class are small. Reference [13] uses fisher score to compare one feature subset with another one in order to find the most discriminating set of feature instances. Reference [14] has used fisher score for online selection of most discriminative set of tracking features. Since ours is a multicamera setup, we need to adapt this fisher score to avoid certain undesirable scenarios from affecting the final discriminant score. Constraints we lay on fisher score are as follows.

1) In a multicamera tracking problem, the discriminating power of tracklet features should be measured across cameras and not intra camera. Thus, in (19), instead of calculating the mean over all tracklets over both cameras, we calculate mean only on the camera with candidate matching tracklets.
2) Online descriptor weight w_f of the fth feature obtained while calculating GMS specifies the robustness of that feature. While calculating mean and the variance of the fth feature of the ith tracklet, we use w_f to weight that mean and variance of the fth feature to specify the influence of such features on fisher score.

Let $\mathbb{k}$ be the set of all features, individual fisher score for any feature $f_k \forall k \in [1 \ldots |\mathbb{k}|]$ is calculated as

$$\delta(f_k) = \frac{\sum_{i=1}^{N} w_{f_k}\left(\mu_{if_k} - \mu_{f_k}^{C^r}\right)^2}{\sum_{i=1}^{N} w_{f_k}\left(\rho_{if_k}^2\right)} \quad (19)$$

where μ_{if_k} and ρ_{if_k} are the mean and the variance of the kth GMS feature of the ith tracklet, N is the number of tracklets in camera C^l, w_{f_k} is the descriptor similarity weight of the kth feature, and $\mu_{f_k}^{C^r}$ is the mean of the kth GMS feature of overall candidate tracklets belonging to complementary pair of camera C^r.

Normalized fisher score for the kth GMS feature is calculated as $\delta'(f_k)$

$$\delta'(f_k) = \frac{\delta(f_k)}{\sum_{z=1}^{|F|} \delta(f_z)}. \quad (20)$$

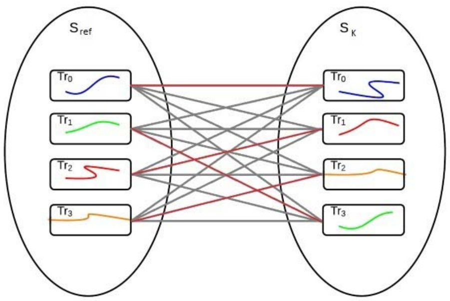

Fig. 6. Associations of each trajectory after Hungarian algorithm.

1) Consistent Discriminancy of Tracklet Features: An individual consistency score is obtained for each feature f_k in GMS metric over the entire tracklet (Tr_i) as

$$\upsilon(f_k, \mathrm{Tr}_i) = \sqrt{\frac{\sum_{t=0}^{n_k}\left(f_k(\mathrm{TO}_t^i) - \overline{f_k(\mathrm{Tr}_i)}\right)^2}{n_k}} \tag{21}$$

where $f_k(\mathrm{TO}_t^i)$ is the kth feature extracted from the ith tracked object TO^i at time t, $\overline{f_k(\mathrm{Tr}_i)}$ is the kth feature mean over trajectory of tracked object TO^i, and n_k is the total number of detections.

Normalized individual consistency score $\upsilon'(f_k, \mathrm{Tr}_i)$ of the kth feature $\upsilon'(f_k, \mathrm{Tr}_i)$ is calculated as

$$\upsilon'(f_k, \mathrm{Tr}_i) = \frac{\upsilon(f_k)}{\sum_{z=1}^{|F|}\upsilon(f_z)}. \tag{22}$$

GLCDS of features on an entire tracklet is calculated by taking square root of sum of weighted consistency score of individual features over a tracklet Tr_i

$$\lambda_m(\mathrm{Tr}_i) = \frac{\sqrt{\delta'(f_1)\cdot\upsilon'(f_1, \mathrm{Tr}_i)^2 + .. + \delta'(f_{|F|})\cdot\upsilon'(f_{|F|}, \mathrm{Tr}_i)^2}}{|F|}. \tag{23}$$

E. Hungarian Algorithm

The task at hand is finding the maximum matching of G. Formally, maximum matching is defined as a matching with the largest possible number of edges; it is globally optimal. The goal is to find an optimal assignment, i.e., find the maximum matching in G. We apply the Hungarian algorithm defined in [15] given the cost matrix built with the A_{ij} values. After applying the Hungarian algorithm to matrix A, we get the maximum matching as shown in Fig. 6. The red lines specify the established associations between tracklets across cameras as a result of the Hungarian algorithm.

VIII. Trajectory Fusion

Trajectory confidence score R_{TO} can be intuitively interpreted as how well tracklets' fusion from individual cameras can match the original trajectory of target. We calculate individual tracklets confidence based on the following.

1) *Length:* Long trajectories are more reliable, and therefore trajectories below a handpicked short length are unreliable.
2) *Geometric Coherence Score:* Assuming that the variation of tracklet features follow a Gaussian distribution, the coherence score is calculated as follows.
   From (6), TO_t^i is the position of object TO^i at time t and TO_{t-1}^i is previous position of object TO^i. The coherence score ϖ is defined as
   $$\varpi = \frac{1}{\sqrt{2\pi\sigma_i^2}} e^{-\frac{(d_i-\mu_i)^2}{2\sigma_i^2}} \tag{24}$$
   where d_i is the 2D distance between TO^i and TO_{t-1}^i, μ_i and σ_i are, respectively, the mean and standard deviation of frame-to-frame distance distribution formed by a set of positions of object TO^i.
3) *Appearance Coherence Score:* Similar to geometric coherence score, but here we account for an array of appearance features. Here d_i represents the distance between feature descriptors at TO^i and TO_{t-1}^i

Confidence score R_{TO} of a tracklet is the mean of all the above coherence scores.

As part of the fusion task, a merged trajectory with the information coming from both views is built. To fuse two trajectories coming from two different cameras at a time t, e.g., $\mathrm{Tr}_i \in S_l$ with $0 < i < N$ and $\mathrm{Tr}_j \in S_r$ with $0 < j < N$ into a global one $\mathrm{Tr}_{Gi,Gj}$, we apply an adaptive weighting method as

$$\mathrm{Tr}_{Gi,Gj}(t) = \begin{cases} \psi_1 \mathrm{Tr}_i^{Cl}(t) + \psi_2 \mathrm{Tr}_j^{Cr}(t) & \text{if } \mathrm{Tr}_i^{Cl}(t), \mathrm{Tr}_j^{Cr}(t) \\ \quad \text{overlap over time t} & \\ \mathrm{Tr}_i^{Cl}(t) & \text{if only } \mathrm{Tr}_i^{Cl}(t) \text{ exists at time } t \\ \mathrm{Tr}_j^{Cr}(t) & \text{if only } \mathrm{Tr}_j^{Cr}(t) \text{ exists at time } t \end{cases} \tag{25}$$

where ψ_1 and ψ_2 are the weights calculated as in (26). Each tracked object has a reliability attribute R_{TO} with values [0, 1], and the weighed function is defined in terms of its R_{TO} value as

$$\psi_1 = \frac{R_{\mathrm{TO}_i}}{R_{\mathrm{TO}_i} + R_{\mathrm{TO}_j}} \quad \psi_2 = \frac{R_{\mathrm{TO}_j}}{R_{\mathrm{TO}_i} + R_{\mathrm{TO}_j}} \tag{26}$$

where R_{TO_i} and R_{TO_j} are the reliability attributes of tracked object from camera C^l and C^r, respectively.

The fused trajectory is not smooth. In order to get a better and smoothed one, we apply a simple moving average technique (also called moving mean).

IX. Evaluation

Our RGB approach is evaluated on publicly available PETS2009 data set [16]. We choose to evaluate on View 1, View 3, View 5, and View 7 in S2.L1 scenario. There is one static occlusion in View 1, namely, a pole with display board, and View 3 is quite challenging as a tree occupies significant area in the right side of video. Also there is substantial color tone variation between the views, making it hard for color-based cues. For this reason, most of the methods avoid this combination of view. To show the effectiveness of GLCDS, we take up this challenging view as it more resembles real-world scenario.

TABLE I

RESULT COMPARISON IN PERCENTAGE. THE BEST CONFIGURATION OF OUR SYSTEM IS MARKED WITH THE BLUE BACKGROUND

Dataset	Approach	Cam ID	MOTA	MOTP	MT	PT	ML	IDS
PETS 2009	Berclaz et al	1,3,5,6,8	82	56	-	-	-	-
	Leal-Taixe et al	1,5	76	60	-	-	-	-
	Leal-Taixe et al	1,5,6	71.4	53.4	-	-	-	-
	Murray Evans et al	-	63	55	-	-	-	-
	Martin Hofmann	1,5	99.4	82.9	100	0	0	1
	Martin Hofmann	1,5,7	99.4	83	100	0	0	2
	Our approach (C3)	1,3,5,7	86	77.2	93.7	4.6	1.7	0
	Our approach (C2)	1,3,5,7	86.8	77.4	95	2.8	2.2	2
	Our approach (C1)	1,3,5,7	84.3	73.1	90.1	7	3.1	2
	Our approach (C4)	1,3	90.3	80.2	97.1	3	0	0
	Our approach (C4)	1,3,5	92.2	80.8	97.9	2.1	0	0
	Our approach (C4)	1,3,5,7	92.7	81	99	1	0	0

For evaluating our work, we use the following metrics: CLEAR [17] metrics, namely, multiple object tracking accuracy (MOTA) and multiple object tracking precision (MOTP), identity switches (IDS), track fragments, mostly tracked (MT), partly tracked (PT), and mostly lost (ML) from [18].

Table I summarizes comparison between our method and other multicamera approaches on PETS2009 data set. Unlike other methods that use heavy computation and optimization for best results as a tradeoff over real-time performance, our objective was to make the algorithm more real time making minimal sacrifice on the accuracy. This is achieved as our method uses computationally efficient and in-complex optimization technique with dynamic feature ranking and election for an effective ensemble. We use buffer frame size = 20 frames in a temporal sliding window pattern to be able to perform association and fusion online.

We experiment our method with four different system configurations:

1) *C1:* without online learnt feature ensemble selection (GLCDS based);
2) *C2:* without online learnt tracklet appearance models;
3) *C3:* without locality-based features;
4) *C4:* with full configuration.

The evaluation results of each configuration ($C1$–$C4$) show us how much impact each part has on the proposed method. $C4$ is our entire system with fully loaded configuration and is expected to improve the performance to maximum. From Table I, we can see that the absence of GLCDS and online appearance models has introduced the only ML entry among the pool of configurations symbolizing the significance of online learnt feature ensemble. Configurations $C1$ and $C2$ produce IDS stressing on the impact of online appearance models on the framework. Since Views 5 and 7 give a closer view at the overlapping area, appearance features from these views play a vital role. $C4$ altogether produces reliable long trajectories, thereby improving fragmentation, ML, and PL, and also suppresses IDS. We can see that our method surpasses the state of the art in IDS and produces more or less similar results on various other metrics while remaining a real-time online approach.

For evaluating on RGB-D data, we select five videos from a private data set, in which participants with Alzheimer disease aged more than 65 years are recruited by the memory center of a collaborating hospital. The clinical protocol asks the participants to undertake a set of physical tasks and instrumental activities of daily living in a hospital observation room furnished with home appliances. Experimental recordings use two RGB-D cameras (Kinect) with 640 × 480 pixels of resolution and nonlinear time synchronization between them. Each pair of videos has two different views of the scene, lateral and frontal, with a maximum amount of two people per view. A sample frame form the video is shown in Fig. 7.

Fig. 7. One of the frames during the evaluation of the RGB-D video.

TABLE II

COMPARISON OF MULTICAMERA RGBD TRACKER VERSUS MONO-CAMERA RGB TRACKER

Camera	Total Trajectories	MT %	PT%	ML%	IDS
Mono-camera	10	40	60	0	1
Multi-camera	10	100	0	0	0

In our data set, doctor trajectory is cut several times because of occlusions. Sometimes, he appears in one camera and sometimes in the other. The merged trajectory keeps the information of both cameras making a good manage of occlusions.

In this video, the mono-camera tracking has bad results for the doctor in the right camera and even worst for the patient in the left camera. But it can be seen that with multicamera approach, we combine the best results for each camera into a global one, and so finally, we have the two tracked objects that appear in the scene with good tracking results. Our multicamera results improve the mono-camera trajectory significantly as shown in Table II.

These experiments reveal that our framework is robust in rectifying the challenges of conventional mono camera tracking and produces consistent trajectories with no IDS.

Our approach had the results benchmarked based on a view (which actually resembles real world) purposefully ignored by all other methods and also produced improvements to the state of the art while being a real-time approach.

System Implementation

As shown in Fig. 1, our system is implemented with parallel programming to handle multiple cameras in a network as multithreads. Time efficiency of multicamera master thread is appreciable as it takes the same time as the turnaround time of individual worker threads. All individual worker node's local geometric information is projected on to the reference camera's world. Local feature extraction, association, and fusion are all done in the reconstructed reference world, and then projected back to reference camera's image plane for evaluation and visualization. Therefore, theoretically, there are no bounds for number of cameras to run in our framework, as the model is very elastic and extensible. But hardware capability might be a bottleneck.

X. Conclusion

We introduced a multicamera multitarget multimodality online tracking framework that associates and fuses trajectories on the grounds of an online learned consistent and discriminant global–local feature ensemble. Our approach's backbone has been feature engineering, and its performance on the data sets demonstrated the importance of dynamically selecting and ranking features that capture and wholly represent the video properties and contents. As a result of our work, we were able to build optimally long complete trajectories by linking and fusing data based on confidence and reliability scores calculated at individual camera level. Using this framework, we achieve highly parallel and effective real-time performance, which is absent in the state-of-the-art methods. Our approach outperforms some existing multicamera tracking and is comparable with state-of-the-art benchmark data sets. Even when coupled with in-complex optimizations to fasten the algorithm, final results show the impact of engineering feature embeddings and their selection on accuracy and real-time performance.

References

[1] M. Hofmann, D. Wolf, and G. Rigoll, "Hypergraphs for joint multi-view reconstruction and multi-object tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 3650–3657.

[2] M. Evans, L. Li, and J. Ferryman, "Suppression of detection ghosts in homography based pedestrian detection," in *Proc. IEEE 9th Int. Conf. Adv. Video Signal-Based Surveill.*, Sep. 2012, pp. 31–36.

[3] M. Taj and A. Cavallaro, "Distributed and decentralized multicamera tracking," *IEEE Signal Process. Mag.*, vol. 28, no. 3, pp. 46–58, May 2011.

[4] L. Zhang, Y. Li, and R. Nevatia, "Global data association for multi-object tracking using network flows," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2008, pp. 1–8.

[5] H. Pirsiavash, D. Ramanan, and C. C. Fowlkes, "Globally-optimal greedy algorithms for tracking a variable number of objects," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2011, pp. 1201–1208.

[6] J. Berclaz, F. Fleuret, and P. Fua, "Multiple object tracking using flow linear programming," in *Proc. 12th IEEE Int. Workshop Perform. Eval. Tracking Surveill. (PETS-Winter)*, Dec. 2009, pp. 1–8.

[7] M. Evans, C. J. Osborne, and J. Ferryman, "Multicamera object detection and tracking with object size estimation," in *Proc. 10th IEEE Int. Conf. Adv. Video Signal Based Surveill. (AVSS)*, Aug. 2013, pp. 177–182.

[8] N. Anjum and A. Cavallaro, "Trajectory association and fusion across partially overlapping cameras," in *Proc. 6th IEEE Int. Conf. Adv. Video Signal Based Surveill. (AVSS)*, Sep. 2009, pp. 201–206.

[9] Y. A. Sheikh and M. Shah, "Trajectory association across multiple airborne cameras," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 2, pp. 361–367, Feb. 2008.

[10] L. Bergroth, H. Hakonen, and T. Raita, "A survey of longest common subsequence algorithms," in *Proc. IEEE 7th Int. Symp. String Process. Inf. Retr. (SPIRE)*, Sep. 2000, pp. 39–48.

[11] A. Kassidas, J. F. MacGregor, and P. A. Taylor, "Synchronization of batch trajectories using dynamic time warping," *AIChE J.*, vol. 44, no. 4, pp. 864–875, 1998.

[12] D. P. Chau, F. Bremond, and M. Thonnat, "A multi-feature tracking algorithm enabling adaptation to context variations," in *Proc. 4th Int. Conf. Imag. Crime Detection Prevention (ICDP)*, Nov. 2011, pp. 1–6. AQ:4

[13] Q. Gu, Z. Li, and J. Han. (2012). "Generalized Fisher score for feature selection." [Online]. Available: https://arxiv.org/abs/1202.3725

[14] R. T. Collins, Y. Liu, and M. Leordeanu, "Online selection of discriminative tracking features," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 10, pp. 1631–1643, Oct. 2005.

[15] H. W. Kuhn, "The Hungarian method for the assignment problem," *Naval Res. Logistics Quart.*, vol. 2, nos. 1–2, pp. 83–97, 1955.

[16] J. Ferryman and A. Ellis, "PETS2010: Dataset and challenge," in *Proc. 7th IEEE Int. Conf. Adv. Video Signal Based Surveill. (AVSS)*, Aug./Sep. 2010, pp. 143–150.

[17] K. Bernardin and R. Stiefelhagen, "Evaluating multiple object tracking performance: The CLEAR MOT metrics," *EURASIP J. Image Video Process.*, vol. 2008, p. 246309, Dec. 2008. AQ:5

[18] C.-H. Kuo and R. Nevatia, "How does person identity recognition help multi-person tracking?" in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2011, pp. 1217–1224.

[19] Z. Wu, N. I. Hristov, T. L. Hedrick, T. H. Kunz, and M. Betke, "Tracking a large number of objects from multiple views," in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, Sep./Oct. 2009, pp. 1546–1553.

[20] C. Feremans, M. Labbé, and G. Laporte, "Generalized network design problems," *Eur. J. Oper. Res.*, vol. 148, no. 1, pp. 1–13, 2003.

Kanishka Nithin received the master's degree in computer vision and image processing from Amrita School of Engineering, India, in 2015. AQ:6

He has been with the STARS team, INRIA Sophia Antipolis, Sophia Antipolis, France, since 2015, where he is currently a Pre-Ph.D. Researcher, involved in multicamera video surveillance. His research interests include semantic video analytics, object recognition, visual slam, and deep learning.

François Brémond received the Ph.D. degree in video understanding from INRIA Sophia Antipolis, Sophia Antipolis, France, in 1997 and the HDR degree in scene understanding from University of Nice Sophia Antipolis, Nice, France, in 2007.

He was a Post-Doctoral Researcher with University of Southern California, Los Angeles, CA, USA, where he was involved in the interpretation of videos taken from unmanned airborne vehicles. He created the STARS team in 2012. He is currently the Research Director with INRIA Sophia Antipolis. He has been conducting research in video understanding at Sophia-Antipolis since 1993. He has authored or co-authored over 140 scientific papers published in international journals and conferences in video understanding.

Dr. Brémond is a Handling Editor of MVA and a Reviewer for several international journals such as *Computer Vision and Image Understanding*, *International Journal of Pattern Recognition and Artificial Intelligence*, *International Journal of Human-Computer Studies*, IEEE Transactions on Pattern Analysis and Machine Intelligence, *Artificial Intelligence Journal*, and *Eurasip Journal on Advances in Signal Processing* and conferences such as CVPR, ICCV, AVSS, VS, and ICVS. He has supervised or co-supervised 13 Ph.D. theses. He is an EC INFSO and French ANR Expert for reviewing projects. AQ:7